7目录、共享、保护

2019年1月6日 18:30

-
- 文件目录

1. 目录管理的要求：
   1) 按名存取：最基本的功能，以后提供文件名，系统返回存储位置
   2) 高速检索：大中型文件系统的目标
   3) 文件共享：多用户系统的目标，一般通过提供共享副本实现
   4) 允许文件重名：指对不同用户可以对不同文件采用相同名

一. 文件控制块和索引结点

   1) 文件控制块的有序集合即为文件目录
   2) 文件目录也可视作文件：目录文件

1. 文件控制块FCB
   1) 基本信息类
      (1) 文件名：唯一标识
      (2) 文件物理位置：设备名、起始盘块号、占用盘块数或字节数
      (3) 文件逻辑结构：指示文件是否定长，是否有结构、记录数
      (4) 文件的物理结构，指示文件是顺序文件、链接式文件或索引文件
   2) 存取控制信息类：文件主、核准用户、一般用户的不同存取权限
   3) 使用信息类：建立、修改时间、使用信息（使用进程数，锁，是否修改）
   4) 

图 6-15 MS-DOS 的文件控制块

2. 索引结点
   1) 索引结点的引入：查找目录通常需要将存放目录的第一个盘块的目录调入内存，逐个比较，找不到还需调入下一盘块，平均需要总盘块数N+1/2次
      (1) 实际上检索时只有匹配的文件名及其物理地址是有用的信息
      (2) UNIX的文件描述信息放在索引结点i中
      (3) 

图 6-16 UNIX 的文件目录
   2) 磁盘索引结点
      (1) 文件主标识符：拥有该文件的个人或小组的标识符
      (2) 文件类型：正规文件、目录文件或特别文件
      (3) 文件存取权限：各类用户对该文件的存取权限
      (4) 文件物理地址：每个索引结点中有 13 个地址项 iaddr(0) ~ iaddr(12)，以直接或间接方式给出数据文件所在盘块的编号
      (5) 文件长度：以字节为单位的文件长度

(6) 文件连接计数：表明在本文件系统中所有指向该文件的指针计数

(7) 文件存取时间：本文件最近被进程存取、修改的时间、索引结点最近被修改的时间

3) 内存索引结点：文件打开后，磁盘索引结点拷贝到内存中，便于以后使用，增加了以下内容

(1) 索引结点编号：用于标识内存索引结点

(2) 状态：指示 i 结点是否上锁或被修改

(3) 访问计数：每当有一进程要访问此 i 结点时，++该访问计数，访问完再--

(4) 文件所属文件系统的逻辑设备号

(5) 链接指针：分别指向空闲链表和散列队列的指针

二. 简单的文件目录

1. 单级文件目录：整个文件系统只有一张目录表，每个文件占一项

1) 查找速度慢：评价需顺序查找N/2次

2) 不允许重名：新建文件时，检查不与其他文件名相同

3) 不便于共享：必须用唯一文件名来访问，只适于单用户环境

4)

文件名	物理地址	文件说明	状态位
文件名 1			
文件名 2			
…			

图 6-17 单级目录

2. 两级文件目录：为每个用户单独建一个用户文件目录 UFD(User File Directory)，再在系统中建一个主文件目录 MFD(Master File Directory)，每个用户占一行

(1)

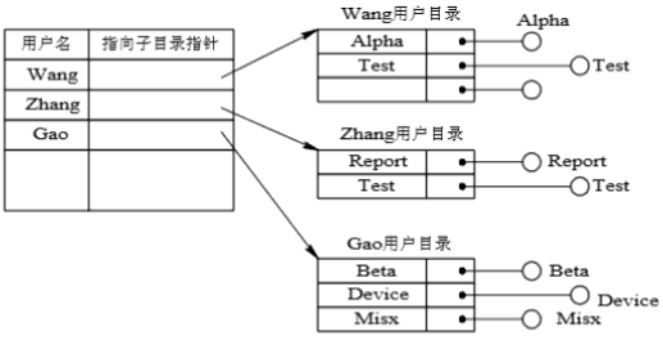


图 6-18 两级目录结构

1) 提高了检索目录的速度： n 个用户各最多为 m 个 目录项，则最多只需检索 n + m 个目录项。但如果是采用单级目录结构，则最多需检索 n × m 个目录项。检索效率提高约 n/2 倍

2) 在不同的用户目录中，可以使用相同的文件名，只有同用户同文件名才需要重命名

3) 不同用户可使用不同文件名访问同一共享文件。在各用户之间完全无关时，这种隔离是一个优点；但当多个用户间要合作，访问对方文件时，这种隔离便成为一个缺点，会使诸用户之间不便于共享文件

三. 树形结构目录Tree-Structured Directory

1. 树形目录：主目录为唯一的根目录，数据文件为树叶，其他子目录为结点

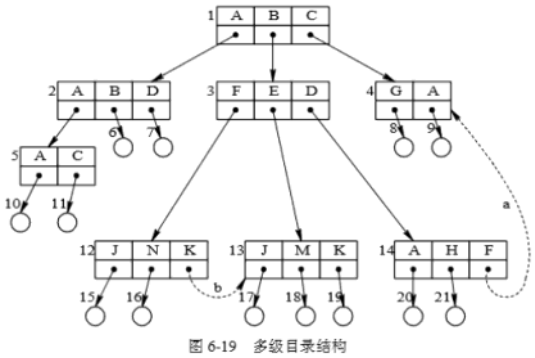
1) 

图 6-19　多级目录结构

2) （圆圈为数据文件，方框为目录文件）

2. 路径名和当前目录
   1) 绝对路径名(absolute path name)：根目录到文件的通路间，每一结点前后都用 / 连接，即构成其唯一路径名，因为从根到任一树叶都只有唯一通路
   2) 当前目录(Current Directory) /工作目录：进程访问范围的父结点
   3) 相对路径名(relative path name)：从当前目录开始的路径名
   4) 查询速度快，层次结构清晰，文件管理保护有效，存取权限易区分
   5) 但磁盘访问次数多
3. 目录操作
   1) 创建目录：用户先创建UFD，再在其中新增不重名的子目录
   2) 删除目录：空目录可直接删，非空目录有两种处理方式：
      (1) 不允许删，只能递归清空其子目录才能删，如MS-DOS
      (2) 允许删，直接将全部子目录和文件清空，较危险
   3) 改变目录：即设置当前目录，默认是到主目录
   4) 移动目录：将文件或子目录在不同父文件间转换
   5) 链接link操作：让文件具有多个父目录，方便共享
   6) 查找：精确匹配文件或局部匹配文件

四. 目录查询技术

1. 线性检索法/顺序检索法：依次读入路径分量名，顺序比较文件名，按索引结点号插索引表，读入其盘块，直到找到文件，任一分量找不到都应返回错误信息

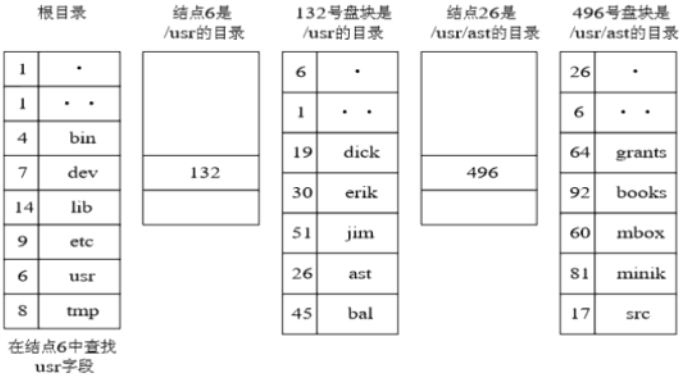
1) 

图 6-20　查找/usr/ast/mbox 的步骤

2. Hash方法：用hash函数求出文件名对应的索引值，再到hash索引目录找
   1) 不适用于使用了通配符的文件名
   2) 冲突：不同文件名可能转换为同一hash值
      (1) 如果hash值在索引表找到目录项为空，说明尚无指定文件
      (2) 若目录项的文件名匹配，说明查找成功

(3) 若非空但不匹配，说明是发生了冲突，因在hash值上加一与目录长度互质的常数，再重新查

-
- 文件共享

1) 可称为绕弯路法和连访法

一. 基于有向无循环图实现文件共享

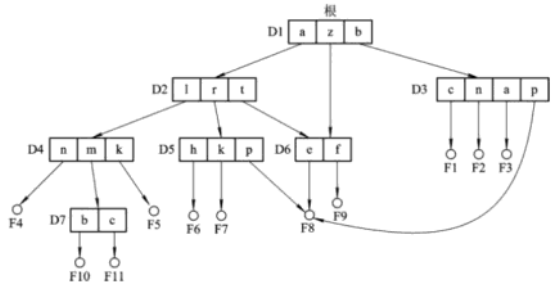
1. 有向无循环图DAG(Directed Acyclic Graph)
   1) 不同用户各自的父目录指向同一文件，多用户以对称方式共享文件
   2) 会产生回路，破坏树形结构
   3) 增加新内容可能要增加新盘块，新增部分对其他用户不可见
   4) 

图7-13 有向无循环图目录层次

2. 利用索引结点：将物理地址等文件属性放在索引结点
   1) 文件目录只含文件名和索引结点号
   2) 新增信息时修改索引结点的内容，使其他用户可见修改
   3) 索引结点中应记录链接计数count，记录共享用户数
   4) count不为0就不能删除文件，计帐系统中，文件主可能要为此“付账”

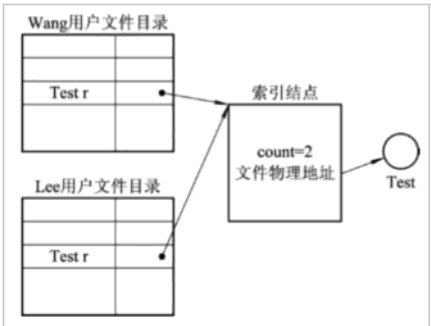


图7-14 基于索引结点的共享方式

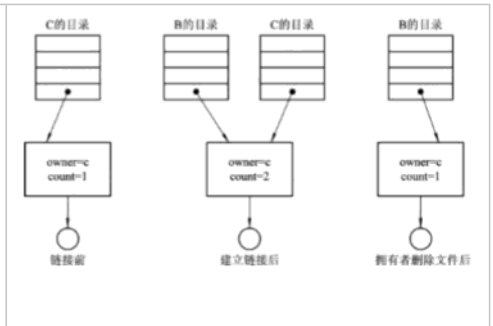


图7-15 进程B链接前后的情况

二. 利用符号链接（软链接）实现文件共享

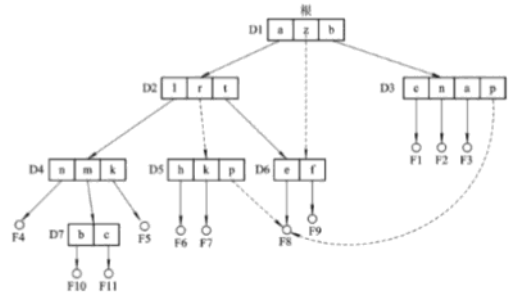
1. 利用符号链接(Symbolic Linking)的基本思想：允许多个父目录存在，除了唯一主父目录外，都是链接父目录，靠符号链接。符号链接视作虚线，则实线部分仍是简单树，删查都很方便
   1) 

图7-16 使用符号链接的目录层次

2. 利用符号链实现共享：在链接父目录中创建link型文件，取名同文件名，内容为文件的路径名
   1) 即新建一个文件，里面写着该文件的目录，在linux中是ln -s建立的
3. 利用符号链实现共享的优点：
   1) 只有文件主拥有索引结点号，删除文件只会导致访问失败并自动删除链接，不会造成悬空指针误操作
   2) 适用于网络链接分布在世界各地的计算机的文件
4. 利用符号链的共享方式存在的问题：

1) 链接访问可能需要多次读盘，开销大
2) 链接本身也是文件，需要额外分配索引结点，浪费空间
3) 遍历traverse文件系统可能反复读到某一链接，使共享文件被反复拷贝

◆

◆ 文件保护

1. 影响文件安全的主要因素：
   1) 人为因素：人们有意无意的行为使数据被破坏或丢失
   2) 系统因素：系统异常导致数据破坏或丢失，如磁盘故障
   3) 自然因素：随时间推移，磁盘上的数据会逐渐消失
2. 为确保文件安全性可采取：
   1) 存取控制权限，防止人为因素
   2) 采取系统容错技术，防止系统因素
   3) 建立后备系统，

一. 保护域(Protection Domain)

1. 访问权(Access right)<对象名，权集>：进程对某对象执行操作的权力
2. 保护域/域：进程拥有某访问权的对象的集合，进程只能在指定域中执行操作

1)

图7-17 三个保护域

3. 进程和域间的静态联系：进程和域之间可以一一对应，即在进程的整个生命期中，可用资源是固定的，这种域称为“静态域”
   1) 进程运行全过程都受限于同一个域，会使赋予进程的访问权超过实际需要
4. 进程和域之间的动态联系方式：一个进程对应多个域，每个阶段不同域
   1) 称其为动态联系方式，需要系统中有保护域切换功能

二. 访问矩阵(Access Matrix)，

1. 访问矩阵：描述系统访问控制的矩阵，行为域，列为对象
   1) 访问权access(i, j)定义了在域Di中执行 的进程能对对象Qj所施加的操作集
2. 具有域切换权的访问矩阵
   1) 将切换能力作为一种权力S
   2) 当且仅当switch∈access(i, j)时，才允许进程从域i切换到域j

3)

域 \ 对象	F_1	F_2	F_3	F_4	F_5	F_6	Printer 1	Plotter 2	域 D_1	域 D_2	域 D_3
域D_1	R	R, W								S	
域D_2			R	R, W, E	R, W		W				S
域D_3						R, W, E	W	W			

图7-19 具有切换权的访问控制矩阵

(1) 如图：域d1可以切换到d2，d2可以到d3，但不能反向切

三. 访问矩阵的修改

1. 拷贝权(Copy Right)：打星号的权可以扩展到同列其他域中

1)

域 \ 对象	F_1	F_2	F_3
D_1	E		W*
D_2	E	R*	E
D_3	E		

(a)

域 \ 对象	F_1	F_2	F_3
D_1	E		W*
D_2	E	R*	E
D_3	E	R	W

(b)

图7-20 具有拷贝权的访问控制矩阵

   2) 如图，d1和d2把W*和R*扩展给了d3，但扩展后没了*
   3) 限制拷贝：拷贝时不降拷贝权*扩展，限制访问权进一步扩展
2. 所有权(Owner Right)：增加或删除O所在列各种权的能力

1)

域 \ 对象	F_1	F_2	F_3
D_1	O, E		W
D_2		R^*, O	R^*, O, W
D_3	E		

(a)

域 \ 对象	F_1	F_2	F_3
D_1	O, E		
D_2		O, R^*, W^*	R^*, O, W
D_3		W	W

(b)

图7-21 带所有权的访问矩阵

2) 如图，各种增删权限都可以做到，甚至O所在行本身还可以加一个W*

3. 控制权(Control Right)：Control该行各种权限的增删

1)

域 \ 对象	F_1	F_2	F_3	F_4	F_5	F_6	Printer 1	Plotter 2	域 D_1	域 D_2	域 D_3
域D_1	R	R, W									
域D_2			R	R, W, E	R, W		W				Control
域D_3						R, E	W	W			

图7-22 具有控制权的访问矩阵

2) 如图：d2可以修改d3的权限

四. 访问矩阵的实现

1. 访问控制表(Access Control List)：为每列对象建成一条ACL
   1) 通过不记录空项，减少占用的存储空间，提供查找速度
   2) 一般是每个文件一条ACL，记录有权限的用户进程
   3) 可用于定义缺省访问权集，如果默认ACL找不到该进程想要的权限，才去对象的ACL中查找
2. 访问权限capabilities表：为每行的域都建一张表，每行为对对应对象的权限

1)

	类 型	权力	对 象
0	文件	R—	指向文件 3 的指针
1	文件	RWE	指向文件 4 的指针
2	文件	RW-	指向文件 5 的指针
3	打印机	-W-	指向打印机 1 的指针

图7-23 访问权限表

2) 仅当权限表安全时，保护的对象才可能安全，因此该表只能被通过合法性检查的程序访问

3. 访问控制表和权限表一般是同时存在的，先检查控制表，无权访问便拒绝，并构成异常，有权则建立一权限，连接到该进程，之后，进程便可用该权限访问对象

i.

ii.

iii.

iv.

v. ---------我是底线-------